PATENT
2557-000172/US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	New Application	Group Art Unit:	Unknown
Filing Date:	August 27, 2003	Examiner:	Unknown
Applicants:	Jin-Cheon KIM	Conf. No.:	Unknown
Title:	CACHE CONTROLLER COMPUTER SYSTEM AND METHOD FOR PROGRAM RECOMPILATION		

PRIORITY LETTER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

August, 27, 2003

Dear Sirs:

Pursuant to the provisions of 35 U.S.C. 119, enclosed is/are a certified copy of the following priority document(s).

Application No.	Date Filed	Country
2003-0007414	February 6, 2003	Korea

In support of Applicant's priority claim, please enter this document into the file.

Respectfully submitted,

HARNESS, DICKEY, & PIERCE, P.L.C.

By _____
John A. Castellano, Reg. No. 35,094

P.O. Box 8910
Reston, Virginia 20195
(703) 668-8000

JAC:bmd

Enclosure

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2003-0007414
Application Number

출 원 년 월 일 : 2003년 02월 06일
Date of Application FEB 06, 2003

출 원 인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2003 년 05 월 30 일

특 허 청
COMMISSIONER

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0021
【제출일자】	2003.02.06
【국제특허분류】	G06F
【발명의 명칭】	쓰레드 바이너리 컴파일러에 의하여 프로그램에서 여러 개의 쓰레드를 다이내믹하게 추출하는 컴퓨터 시스템 및 그 동시 다중 쓰레딩 방법
【발명의 영문명칭】	Computer system providing for recompiling a program and cxtracting threads dynamically by a thread binary compiler and Simultaneous Multithreading method thereof
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	정상빈
【대리인코드】	9-1998-000541-1
【포괄위임등록번호】	2003-003437-4
【발명자】	
【성명의 국문표기】	김진천
【성명의 영문표기】	KIM,Jin Cheon
【주민등록번호】	700306-1029427
【우편번호】	449-846
【주소】	경기도 용인시 수지읍 풍덕천리 1167 진산마을 삼성수지5차아파트 52 6-1401호
【국적】	KR
【심사청구】	청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인
이영필 (인) 대리인
정상빈 (인)

【수수료】

【기본출원료】	20	면	29,000 원
【가산출원료】	3	면	3,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	20	항	749,000 원
【합계】	781,000 원		

【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】

【요약】

쓰레드 바이너리 컴파일러에 의하여 프로그램에서 여러 개의 쓰레드를 다이내믹하게 추출하는 컴퓨터 시스템 및 그 동시 다중 쓰레딩 방법이 개시된다. 상기 컴퓨터 시스템은, 캐쉬에 소정 쓰레드 바이너리 컴파일러를 로딩하고, 메인 메모리에 로딩된 프로그램을 캐쉬에 로딩할 때마다, 소정 쓰레드 바이너리 컴파일러에 의한 프로그램의 리컴파일링을 통하여 프로그램이 다수개의 쓰레드들로 분산되는 리컴파일드 프로그램으로 로딩되도록 다이내믹하게 제어한다. 따라서, 유저가 직접 용이하고 자유롭게 바이너리를 리컴파일링하는 것에 의하여 컴퓨터 시스템의 성능을 향상시킬 수 있는 효과가 있다.

【대표도】

도 1

【명세서】

【발명의 명칭】

쓰레드 바이너리 컴파일러에 의하여 프로그램에서 여러 개의 쓰레드를 다이내믹하게 추출하는 컴퓨터 시스템 및 그 동시 다중 쓰레딩 방법{Computer system providing for recompiling a program and cxtracting threads dynamically by a thread binary compiler and Simultaneous Multithreading method thereof}

【도면의 간단한 설명】

본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 간단한 설명이 제공된다.

도 1은 본 발명의 일실시예에 따른 컴퓨터 시스템의 블록도이다.

도 2는 도 1의 컴퓨터 시스템의 동작 설명을 위한 흐름도이다.

도 3은 동시 다중 쓰레딩(SMT) 프로세서의 기능 설명을 위한 도면이다.

도 4는 쓰레드 바이너리 컴파일러(TBC)의 인스트럭션 셋 아키텍쳐(ISA) 컨버젼(conversion) 기능을 설명하기 위한 도면이다.

도 5는 쓰레드 바이너리 컴파일러(TBC)의 프로그램 처리 기능을 설명하기 위한 도면이다.

도 6은 쓰레드 바이너리 컴파일러(TBC)의 다수 프로그램 처리 기능을 설명하기 위한 도면이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<8> 본 발명은 컴퓨터 시스템의 CPU(중앙연산처리장치)에 관한 것으로, 특히 컴퓨터 시스템의 동시 다중 쓰레딩(Simultaneous Multithreading)(이하 "SMT"라 약칭함)에 관한 것이다.

<9> CPU(중앙연산처리장치) 등에서의 수퍼 스칼라(Superscalar) 파이프 라인 구조는, 한 클락 사이클 동안에 동시에 여러 개의 명령어(instruction)를 실행함으로써 CPU(중앙연산처리장치)의 성능을 높이는 방법이다. 그러나, 수퍼 스칼라(Superscalar) 파이프 라인 구조에서는 한 클락 사이클 동안에 동시에 여러 개의 명령어(instruction)가 실행될 수 있음에도 불구하고, 데이터/리소스/콘트롤 디펜던시(data dependency), 캐쉬 미스(cache miss) 등의 문제로 인하여 명령어(instruction)의 실행이 불가능한 구간, 즉, 버티칼 웨이스트(vertical waste)와 호리존탈 웨이스트(horizontal waste)가 발생한다.

<10> 이에 반하여, 동시 다중 쓰레딩(SMT) 방법은 한 클락 사이클 동안에 동시에 여러 쓰레드들(threads)이 동시에 존재하도록 하고, 쓰레드들(threads) 각각에 존재하는 명령어들(instructions)이 동시에 실행되도록 하며, 특히 위와 같은 버티컬 또는 호리존탈 웨이스트(waste) 구간에서도 실행되도록 하여, 웨이스트(waste) 구간을 최소화함으로써 CPU(중앙연산처리장치)의 성능을 향상시키는 방법이다. 쓰레드(thread)는 한 프로세스 내의 여러 가지 서로 다른 콘트롤 포인트(control point)나 서로 다른 실행 경로(execution path)로 정의될 수 있고, 또는 서로 다른 프로그램으로도 정의될 수 있다.

따라서, 동시 다중 쓰레딩(SMT)은 여러 쓰레드들(threads) 각각의 명령어들(instructions)이 동시에 수행되도록 하여 명령어(instruction) 실행의 쓰루풋(throughput)을 높이는 방법이다. 동시 다중 쓰레딩(SMT) 방법에 관한 몇 가지 알고리즘에 대하여, 미국 특허, "US6470443B1"에 잘 나타나 있다.

<11> 따라서, 쓰레드(thread)를 어떻게 발생시켜, 동시 다중 쓰레딩(SMT) 프로세서에서 쓰레드(thread)를 어떻게 처리할 것인 가는 컴퓨터 시스템의 성능에 중요한 요소이다. 쓰레드(thread)를 발생시키는 방법은 입력되는 프로그램을 몇 개의 쓰레드들(threads)로 분산시켜 공급할 것인가의 문제로서, 하드웨어 또는 소프트웨어적으로 구현될 수 있다.

<12> 하드웨어적으로 쓰레드들(threads)을 분산시키는 방법은, 하드웨어에서 프로그램 시퀀스(sequence)를 검출하여 입력되는 프로그램을 여러 개의 쓰레드들(threads)로 다이내믹하게 분산시키는 방법으로서, DMT(Dynamic Multithreading) 방법이라고 한다. 그러나, 이와 같이 하드웨어적으로 쓰레드들(threads)을 분산시키는 방법은 상당히 많이 필요한 하드웨어 장치로 인하여, 회로 및 로직의 복잡도가 증가하여 실제 구현이 상당히 어렵다는 문제점이 있다.

<13> 소프트웨어적으로 쓰레드들(threads)을 분산시키는 방법은, 컴파일러(compiler)에서 입력되는 프로그램을 여러 개의 쓰레드들(threads)로 스테틱(static)하게 분산시키는 방법이다. 이와 같은 방법은, 유저(user)가 소정 응용 프로그램에 의하여 소스(source) 코드를 컴파일링(compiling)하거나, 기존의 바이

너리(binary)를 직접 리컴파일링(recompiling)하는 방법으로 가능하다. 그러나, 이와 같이 스테틱(static)하게 소프트웨어적으로 쓰레드들(threads)을 분산시키는 방법에서는, 유저(user)가 바이너리(binary)를 자유롭게 사용하는데 한계가 있고, 또한 유저(user)가 바이너리(binary)를 분석하여 직접 리컴파일링(recompiling) 하기에는 부적합하다는 문제점이 있다.

【발명이 이루고자 하는 기술적 과제】

<14> 따라서, 본 발명이 이루고자하는 기술적 과제는, 쓰레드 바이너리 컴파일러(thread binary compiler)(이하 "TBC"로 약칭함)에 의하여 프로그램, 즉, 타겟 바이너리(target binary)에서 여러 개의 쓰레드들(thread)을 다이내믹하게 추출하여 네이티브 바이너리(native binary)를 자동으로 만들 수 있는 컴퓨터 시스템을 제공하는 데 있다.

<15> 본 발명이 이루고자하는 다른 기술적 과제는, 쓰레드 바이너리 컴파일러(TBC)에 의하여 프로그램, 즉, 타겟 바이너리(target binary)에서 여러 개의 쓰레드들(thread)을 다이내믹하게 추출하여 네이티브 바이너리(native binary)를 자동으로 만들 수 있는 컴퓨터 시스템의 동시 다중 쓰레딩(SMT) 방법을 제공하는 데 있다.

【발명의 구성 및 작용】

<16> 상기의 기술적 과제를 달성하기 위한 본 발명에 따른 컴퓨터 시스템은, 하드 디스크(hard disk)/플래쉬(flash) 메모리, 메인 메모리(main memory), 캐쉬(cache), 캐쉬(cache) 콘트롤러, 및 동시 다중 쓰레딩(SMT) 프로세서를 구비한다.

<17> 상기 캐쉬(cache) 콘트롤러는 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리로부터 상기 메인 메모리(main memory)에 로딩된 프로그램을 상기 캐쉬(cache)에 로딩할

때마다, 소정 쓰레드 바이너리 컴파일러(TBC)에 의한 상기 프로그램의 리컴파일링(recompiling)을 통하여 상기 프로그램이 다수개의 쓰레드들(threads)로 분산되는 리컴파일드(recompiled) 프로그램으로 로딩되도록 다이내믹하게 제어한다.

<18> 상기 동시 다중 쓰레딩(SMT) 프로세서는 상기 쓰레드들(threads) 각각의 명령어들(instructions)이 호리존탈(horizontal)/버티칼(vertical) 웨이스트가 최소화되어 동시에 실행되도록 처리한다.

<19> 상기 캐쉬(cache)는, 다수개의 프로그램으로부터 리컴파일링(recompiling)된 다수개의 리컴파일드(recompiled) 프로그램을 저장하는 것을 특징으로 한다.

<20> 상기 리컴파일드(recompiled) 프로그램은, 상기 메인 메모리(main memory)에 로딩된 프로그램과 다른 구조의 인스트럭션 셋 아키텍쳐(instruction set architecture)(이하 "ISA"로 약칭함)이며, 인스트럭션 레벨(instruction level)까지 리컴파일링(recompiling)된 것을 특징으로 한다.

<21> 상기 소정 쓰레드 바이너리 컴파일러(TBC)는, 상기 메인 메모리(main memory)에 로딩되어 상주하거나 쓰레드 바이너리 리컴파일링 동작이 일어날 때마다 상기 캐쉬(cache)에 로딩되어 운영되는 것을 특징으로 한다.

<22> 상기 실행이 종료된 상기 리컴파일드(recompiled) 프로그램은, 삭제되거나 상기 메인 메모리(main memory)에 저장되는 것을 특징으로 한다.

<23> 상기 메인 메모리(main memory)에 저장된 상기 리컴파일드(recompiled) 프로그램은, 상기 메인 메모리(main memory)의 용량 초과시 삭제되거나, 리컴파일링 필

요 없이 재사용 될 수 있도록 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리에 저장되는 것을 특징으로 한다.

<24> 상기의 다른 기술적 과제를 달성하기 위한 본 발명에 따른 컴퓨터 시스템의 동시 다중 쓰레딩(SMT) 방법은, 하드 디스크(hard disk)/플래쉬(flash) 메모리, 메인 메모리(main memory), 캐쉬(cache), 캐쉬(cache) 콘트롤러, 및 동시 다중 쓰레딩(SMT) 프로세서를 구비하는 컴퓨터 시스템의 동시 다중 쓰레딩(SMT) 방법에 있어서, 다음과 같은 단계를 구비한다.

<25> 즉, 본 발명에 따른 컴퓨터 시스템의 동시 다중 쓰레딩(SMT) 방법에서는, 먼저, 상기 컴퓨터 시스템이 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리로부터 상기 메인 메모리(main memory)에 프로그램을 로딩한다. 상기 캐쉬(cache) 콘트롤러는 상기 캐쉬(cache)에 소정 쓰레드 바이너리 컴파일러(TBC)를 로딩하고, 상기 메인 메모리(main memory)에 로딩된 프로그램을 상기 캐쉬(cache)에 로딩할 때마다, 상기 소정 쓰레드 바이너리 컴파일러(TBC)에 의한 상기 프로그램의 리컴파일링(recompiling)을 통하여 상기 프로그램이 다수개의 쓰레드들(threads)로 분산되는 리컴파일드(recompiled) 프로그램으로 로딩되도록 다이내믹하게 제어한다. 이에 따라, 상기 동시 다중 쓰레딩(SMT) 프로세서는 상기 쓰레드들(threads) 각각의 명령어들(instructions)이 호리존탈(horizontal)/버티칼(vertical) 웨이스트가 최소화되어 동시에 실행되도록 처리한다.

<26> 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.

<27> 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 각 도면에 제시된 동일한 참조부호는 동일한 부재를 나타낸다.

<28> 도 1은 본 발명의 일실시예에 따른 컴퓨터 시스템의 블록도이다.

<29> 도 1을 참조하면, 본 발명의 일실시예에 따른 컴퓨터 시스템은, 하드 디스크(hard disk)/플래쉬(flash) 메모리(110), 메인 메모리(main memory)(120), 캐쉬(cache)(130), 캐쉬(cache) 콘트롤러(140), 및 동시 다중 쓰레딩(SMT) 프로세서(150)를 구비한다. 동시 다중 쓰레딩(SMT) 프로세서(150)는 일반적인 CPU(중앙연산처리장치)에 해당한다.

<30> 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)는 일반적인 컴퓨터 상의 대용량 메모리로서, 리컴파일드(recompiled) 프로그램을 저장하는 리컴파일드(recompiled) 프로그램 DB(113)와 기본 프로그램이나 응용 프로그램 등을 저장하는 프로그램 DB(115)로 구성될 수 있다.

<31> 상기 메인 메모리(main memory)(120)는, 상기 동시 다중 쓰레딩(SMT) 프로세서(150)와 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110) 사이에 탑재되는 저장 장치로서, 상기 동시 다중 쓰레딩(SMT) 프로세서(150)가 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)로부터 데이터를 읽고 쓰는 시간을 줄여주어 컴퓨터의 동작 스피드를 높인다.

<32> 상기 캐쉬(cache)(130)는, 일반적으로 상기 동시 다중 쓰레딩(SMT) 프로세서(150)와 상기 메인 메모리(main memory)(120) 사이에 탑재되는 임시 저장 장치로서, 상기 동시 다중 쓰레딩(SMT) 프로세서(150)가 상기 메인 메모리(main memory)(120)에 접근하여 데이터를 읽거나 기록할 때, 그 내용의 사본을 상기 메인 메모리(main memory)(120)의

주소와 함께 저장시켜, 상기 동시 다중 쓰레딩(SMT) 프로세서(150)가 상기 메인 메모리(main memory)(120)로부터 데이터를 읽고 쓰는 시간을 줄여주어 컴퓨터의 동작 스피드를 높인다.

<33> 상기 캐쉬(cache) 콘트롤러(140)는 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)로부터 상기 메인 메모리(main memory)(120)에 로딩된 프로그램을 상기 캐쉬(cache)(130)에 로딩할 때마다, 소정 쓰레드 바이너리 컴파일러(TBC)에 의한 상기 프로그램의 리컴파일링(recompiling)을 통하여 상기 프로그램이 다수개의 쓰레드들(threads)로 분산되는 리컴파일드(recompiled) 프로그램으로 로딩되도록 다이내믹하게 제어한다.

<34> 상기 동시 다중 쓰레딩(SMT) 프로세서(150)는 상기 쓰레드들(threads) 각각의 명령어들(instructions)이 호리존탈(horizontal)/버티칼(vertical) 웨이스트가 최소화되어 동시에 실행되도록 처리한다.

<35> 상기 캐쉬(cache)(130)는, 다수개의 프로그램으로부터 리컴파일링(recompiling)된 다수개의 리컴파일드(recompiled) 프로그램을 저장하는 것을 특징으로 한다.

<36> 상기 리컴파일드(recompiled) 프로그램은, 상기 메인 메모리(main memory)(120)에 로딩된 프로그램과 다른 구조의 인스트럭션 셋 아키텍쳐(ISA)이며, 인스트럭션 레벨(instruction level)까지 리컴파일링(recompiling)된 것을 특징으로 한다.

<37> 상기 소정 쓰레드 바이너리 컴파일러(TBC)는, 상기 메인 메모리(main memory)(120)에 로딩되어 상주하는 소프트웨어이며, 쓰레드 바이너리 리컴파일링 동작이 일어날 때마다 상기 캐쉬(cache)(130)에 로딩되어 운영되는 것을 특징으로 한다.

<38> 상기 실행이 종료된 상기 리컴파일드(recompiled) 프로그램은, 삭제되거나 상기 메인 메모리(main memory)(120)에 저장되는 것을 특징으로 한다.

<39> 상기 메인 메모리(main memory)(120)에 저장된 상기 리컴파일드(recompiled) 프로그램은, 상기 메인 메모리(main memory)(120)의 용량 초과시 삭제되거나, 리컴파일링 필요 없이 재사용 될 수 있도록 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)에 저장되는 것을 특징으로 한다.

<40> 상기한 바와 같은, 본 발명의 일실시예에 따른 컴퓨터 시스템의 동작을 좀더 상세하게 설명한다.

<41> 도 2는 도 1의 컴퓨터 시스템의 동작 설명을 위한 흐름도이다.

<42> 도 2를 참조하면, 하드 디스크(hard disk)/플래쉬(flash) 메모리(110), 메인 메모리(main memory)(120), 캐쉬(cache)(130), 캐쉬(cache) 콘트롤러(140), 및 동시 다중 쓰레딩(SMT) 프로세서(150)를 구비하는 도 1의 컴퓨터 시스템의 동시 다중 쓰레딩(SMT) 동작은 다음과 같다.

<43> 먼저, 상기 컴퓨터 시스템은 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)로부터 상기 메인 메모리(main memory)(120)에 프로그램을 로딩한다(S210). 이와 같은 프로그램의 로딩 동작은, 유저(user)가 컴퓨터를 부팅(booting)하거나 런타임(run time) 동안에 실행 명령을 할 때, 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)에 저장되는 기본 프로그램이나 응용 프로그램이 상기 메인 메모리(main memory)(120)로 이동되어 저장되는 것을 말한다. 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)에 저장되는 기본 프로그램이나 응용 프로그램의 소스 코드(source

code)는 여러 가지 OS(operating system)에서 동작되는 서로 다른 인스트럭션 셋 아키텍처(ISA) 구조를 가질 수 있다.

<44> 한편, 상기 캐쉬(cache) 콘트롤러(140)는 상기 캐쉬(cache)(130)에 소정 쓰레드 바이너리 컴파일러(TBC)를 로딩한다(S220). 이에 따라, 상기 캐쉬(cache) 콘트롤러(140)는 상기 메인 메모리(main memory)(120)에 로딩된 프로그램을 상기 캐쉬(cache)(130)에 로딩할 때마다, 상기 소정 쓰레드 바이너리 컴파일러(TBC)에 의한 상기 프로그램의 리컴파일링(recompiling)을 통하여 상기 프로그램이 다수개의 쓰레드들(threads)로 분산되는 리컴파일드(recompiled) 프로그램으로 로딩되도록 다이내믹하게 제어한다(S230).

<45> 이때, 상기 소정 쓰레드 바이너리 컴파일러(TBC)는, 상기 메인 메모리(main memory)(120)에 로딩되어 상주하거나 쓰레드 바이너리 리컴파일링 동작이 일어날 때마다 상기 캐쉬(cache)(130)에 로딩되어 운영된다.

<46> 다음에, 상기 동시 다중 쓰레딩(SMT) 프로세서(150)는 상기 쓰레드들(threads) 각각의 명령어들(instructions)이 호리존탈(horizontal)/버티칼(vertical) 웨이스트가 최소화되어 동시에 실행되도록 처리한다(S240).

<47> 도 3은 동시 다중 쓰레딩(SMT) 프로세서(150)의 기능 설명을 위한 도면이다.

<48> 도 3을 참조하면, 동시 다중 쓰레딩(SMT) 프로세서(150)는 최적화된 상기 쓰레드들(threads)을 상기 캐쉬(cache)(130)에서 받아, 상기 쓰레드들(threads) 각각의 명령어들(instructions)이 호리존탈(horizontal)/버티칼(vertical) 웨이스트가 최소화되어 동시에 실행되도록 하기 위하여, 먼저, 상기 쓰레드들(threads) 각각에 속하는 프로그램 카운터(program counter)에서 어드레스를 페치(fetch)한다. 페치(fetch)된 어드레

스에 대응하는 명령어들(instructions)은 소정의 명령어(instruction) 캐쉬(cache)(130)에서 리드(read)되고, 디코딩, 레지스터 재명명, 명령어(instruction) 큐(queue) 단계들을 거쳐 실행된다. 동시 다중 쓰레딩(SMT) 프로세서(150)의 일반적인 동작에 대하여는, 미국 특허, "US6470443B1"에 잘 나타나 있다.

<49> 도 4는 쓰레드 바이너리 컴파일러(TBC)의 인스트럭션 셋 아키텍쳐(ISA) 컨버젼(conversion) 기능을 설명하기 위한 도면이다.

<50> 도 4를 참조하면, 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)로부터 상기 메인 메모리(main memory)(120)에 로딩된 기본 프로그램이나 응용 프로그램, 즉, 타겟(target) 바이너리(binary)(소스 코드가 일반적인 컴파일러에 의하여 번역된 머신 코드)는 여러 가지 OS(operating system)에서 동작되는 서로 다른 인스트럭션 셋 아키텍쳐(ISA1) 구조를 가질 수 있다. 이에 따라, 타겟(target) 바이너리(binary)는, 상기 캐쉬(cache) 콘트롤러(140)에 의하여 본 발명의 쓰레드 바이너리 컴파일러(TBC)를 탑재한 컴퓨터의 인스트럭션 셋 아키텍쳐(ISA2) 구조로 리컴파일링(recompiling)되고, 컨버젼(conversion)된 리컴파일드(recompiled) 프로그램, 즉, 네이티브 바이너리(native binary)는 상기 캐쉬(cache)(130)에 로딩된다(S230).

<51> 도 5는 쓰레드 바이너리 컴파일러(TBC)의 프로그램 처리 기능을 설명하기 위한 도면이고, 도 6은 쓰레드 바이너리 컴파일러(TBC)의 다수 프로그램 처리 기능을 설명하기 위한 도면이다.

<52> 도 5 및 도 6을 참조하면, 상기 캐쉬(cache) 콘트롤러(140)는, 쓰레드 바이너리 컴파일러(TBC)를 통하여, 상기 메인 메모리(main memory)(120)에 로딩된 하나의 프로세스(

프로그램)이나 일련의 프로그램들을 리컴파일링(recompiling)하여, 상기 캐쉬(cache)(130)에 하나의 리컴파일드(recompiled) 프로그램이나 다수개의 리컴파일드(recompiled) 프로그램들을 상기 캐쉬(cache)(130)에 로딩시킨다(S230). 하나의 리컴파일드(recompiled) 프로그램은 다수개의 쓰레드들(threads)로 구성되고, 하나의 쓰레드들(thread)은 다수개의 명령어들(instructions)로 구성된다.

<53> 여기서, 상기 쓰레드 바이너리 컴파일러(TBC)에 의한 리컴파일링(recompiling) 작업은, 상기 메인 메모리(main memory)(120)에서 입력되는 프로그램의 시퀀스(sequence)를 스스로 검출하여, 프로그램을 여러 개의 쓰레드들(threads)로 다이내믹하게 분산시키는 것으로서, 이에 의하여 최적화된 동작 스피드를 실현할 수 있도록 하기 위하여, 상기 메인 메모리(main memory)(120)에서 입력되는 프로그램이 인스트럭션 레벨(instruction level)까지 고려되어 리컴파일드(recompiled) 프로그램으로 컨버젼(conversion)된다(S230).

<54> 한편, 상기 실행이 종료된 상기 리컴파일드(recompiled) 프로그램은, 삭제되거나 상기 메인 메모리(main memory)(120)에 저장된다(S250). 상기 메인 메모리(main memory)(120)에 저장된 상기 리컴파일드(recompiled) 프로그램은, 상기 메인 메모리(main memory)(120)의 용량 초과시 삭제되거나, 리컴파일링 필요 없이 재사용 될 수 있도록 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)의 리컴파일드(recompiled) DB(113)에 저장된다(S260).

<55> 위에서 기술한 바와 같이, 본 발명의 일실시예에 따른 컴퓨터 시스템은, 먼저, 상기 하드 디스크(hard disk)/플래쉬(flash) 메모리(110)로부터 상기 메인 메모리(main memory)(120)에 프로그램을 로딩한다(S210). 이에 따라, 상기 캐쉬(cache) 콘트롤러

(140)는 상기 캐쉬(cache)(130)에 소정 쓰레드 바이너리 컴파일러(TBC)를 로딩하고 (S220), 상기 메인 메모리(main memory)(120)에 로딩된 프로그램을 상기 캐쉬 (cache)(130)에 로딩할 때마다, 상기 소정 쓰레드 바이너리 컴파일러(TBC)에 의한 상기 프로그램의 리컴파일링(recompiling)을 통하여 상기 프로그램이 다수개의 쓰레드들 (threads)로 분산되는 리컴파일드(recompiled) 프로그램으로 로딩되도록 다이내믹하게 제어한다(S230). 상기 동시 다중 쓰레딩(SMT) 프로세서(150)는 동시에 존재하는 상기 쓰레드들(threads) 각각의 명령어들(instructions)이 호리존탈(horizontal)/버티칼 (vertical) 웨이스트가 최소화되어 동시에 실행되도록 처리한다(S240).

<56> 이상에서와 같이 도면과 명세서에서 최적 실시예가 개시되었다. 여기서 특정한 용어들이 사용되었으나, 이는 단지 본 발명을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

【발명의 효과】

<57> 상술한 바와 같이 본 발명에 따른 컴퓨터 시스템에서는, 쓰레드 바이너리 컴파일러 (TBC)에 의하여 런타임(run time) 동안 프로그램에서 여러 개의 쓰레드들(threads)을 다이내믹하게 추출하여, 동시 다중 쓰레딩(SMT) 프로세서에 인가한다. 따라서, 유저(user)는 직접 인지하지 못하지만, 용이하고 자유롭게 기존의 바이너리(binary)를 리컴파일링 (recompiling)하는 것에 의하여 컴퓨터 시스템의 성능을 향상시킬 수 있는 효과가 있다.

【특허청구범위】

【청구항 1】

하드 디스크/플래쉬 메모리;

메인 메모리;

캐쉬;

상기 하드 디스크/플래쉬 메모리로부터 상기 메인 메모리에 로딩된 프로그램을 상기 캐쉬에 로딩할 때마다, 소정 쓰레드 바이너리 컴파일러에 의한 상기 프로그램의 리컴파일링을 통하여 상기 프로그램이 다수개의 쓰레드들로 분산되는 리컴파일드 프로그램으로 로딩되도록 다이내믹하게 제어하는 캐쉬 콘트롤러; 및

상기 쓰레드들 각각의 명령어들이 호리존탈/버티칼 웨이스트가 최소화되어 동시에 실행되도록 처리하는 동시 다중 쓰레딩 프로세서를 구비하는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 2】

제 1항에 있어서, 상기 캐쉬는,

다수개의 프로그램으로부터 리컴파일링된 다수개의 리컴파일드 프로그램을 저장하는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 3】

제 1항에 있어서, 상기 리컴파일드 프로그램은,

상기 메인 메모리에 로딩된 프로그램과 다른 구조의 인스트럭션 셋 아키텍쳐인 것을 특징으로 하는 컴퓨터 시스템.

【청구항 4】

제 1항에 있어서, 상기 리컴파일드 프로그램은,

인스트럭션 레벨까지 리컴파일링된 것을 특징으로 하는 컴퓨터 시스템.

【청구항 5】

제 1항에 있어서, 상기 소정 쓰레드 바이너리 컴파일러는,

상기 메인 메모리에 로딩되어 상주하는 소프트웨어인 것을 특징으로 하는 컴퓨터 시스템.

【청구항 6】

제 1항에 있어서, 상기 소정 쓰레드 바이너리 컴파일러는,

쓰레드 바이너리 리컴파일링 동작이 일어날 때마다 상기 캐쉬에 로딩되어 운영되는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 7】

제 1항에 있어서, 상기 실행이 종료된 상기 리컴파일드 프로그램은,

삭제되는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 8】

제 1항에 있어서, 상기 실행이 종료된 상기 리컴파일드 프로그램은,

상기 메인 메모리에 저장되는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 9】

제 8항에 있어서, 상기 메인 메모리에 저장된 상기 리컴파일드 프로그램은,

상기 메인 메모리의 용량 초과시 삭제되는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 10】

제 8항에 있어서, 상기 메인 메모리에 저장된 상기 리컴파일드 프로그램은,

상기 메인 메모리의 용량 초과시 상기 하드 디스크/플래쉬 메모리에 저장되는 것을 특징으로 하는 컴퓨터 시스템.

【청구항 11】

하드 디스크/플래쉬 메모리, 메인 메모리, 캐쉬, 캐쉬 콘트롤러, 및 동시 다중 쓰레딩 프로세서를 구비하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법에 있어서,

상기 컴퓨터 시스템에 의하여, 상기 하드 디스크/플래쉬 메모리로부터 상기 메인 메모리에 프로그램을 로딩하는 단계;

상기 캐쉬 콘트롤러에 의하여, 상기 캐쉬에 소정 쓰레드 바이너리 컴파일러를 로딩하는 단계;

상기 캐쉬 콘트롤러에 의하여, 상기 메인 메모리에 로딩된 프로그램을 상기 캐쉬에 로딩할 때마다, 상기 소정 쓰레드 바이너리 컴파일러에 의한 상기 프로그램의 리컴파일링을 통하여 상기 프로그램이 다수개의 쓰레드들로 분산되는 리컴파일드 프로그램으로 로딩되도록 다이내믹하게 제어하는 단계; 및

상기 동시 다중 쓰레딩 프로세서에 의하여, 상기 쓰레드들 각각의 명령어들이 호리존탈/버티칼 웨이스트가 최소화되어 동시에 실행되도록 처리하는 단계를 구비하는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 12】

제 11항에 있어서, 상기 캐쉬는,

다수개의 프로그램으로부터 리컴파일링된 다수개의 리컴파일드 프로그램을 저장하는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 13】

제 11항에 있어서, 상기 리컴파일드 프로그램은,

상기 메인 메모리에 로딩된 프로그램과 다른 구조의 인스트럭션 셋 아키텍쳐인 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 14】

제 11항에 있어서, 상기 리컴파일드 프로그램은,

인스트럭션 레벨까지 리컴파일링된 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 15】

제 11항에 있어서, 상기 소정 쓰레드 바이너리 컴파일러는,

상기 메인 메모리에 로딩되어 상주하는 소프트웨어인 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 16】

제 11항에 있어서, 상기 소정 쓰레드 바이너리 컴파일러는,

쓰레드 바이너리 리컴파일링 동작이 일어날 때마다 상기 캐쉬에 로딩되어 운영되는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 17】

제 11항에 있어서, 상기 실행이 종료된 상기 리컴파일드 프로그램은,

삭제되는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 18】

제 11항에 있어서, 상기 실행이 종료된 상기 리컴파일드 프로그램은,

상기 메인 메모리에 저장되는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【청구항 19】

제 18항에 있어서, 상기 메인 메모리에 저장된 상기 리컴파일드 프로그램은,

상기 메인 메모리의 용량 초과시 삭제되는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

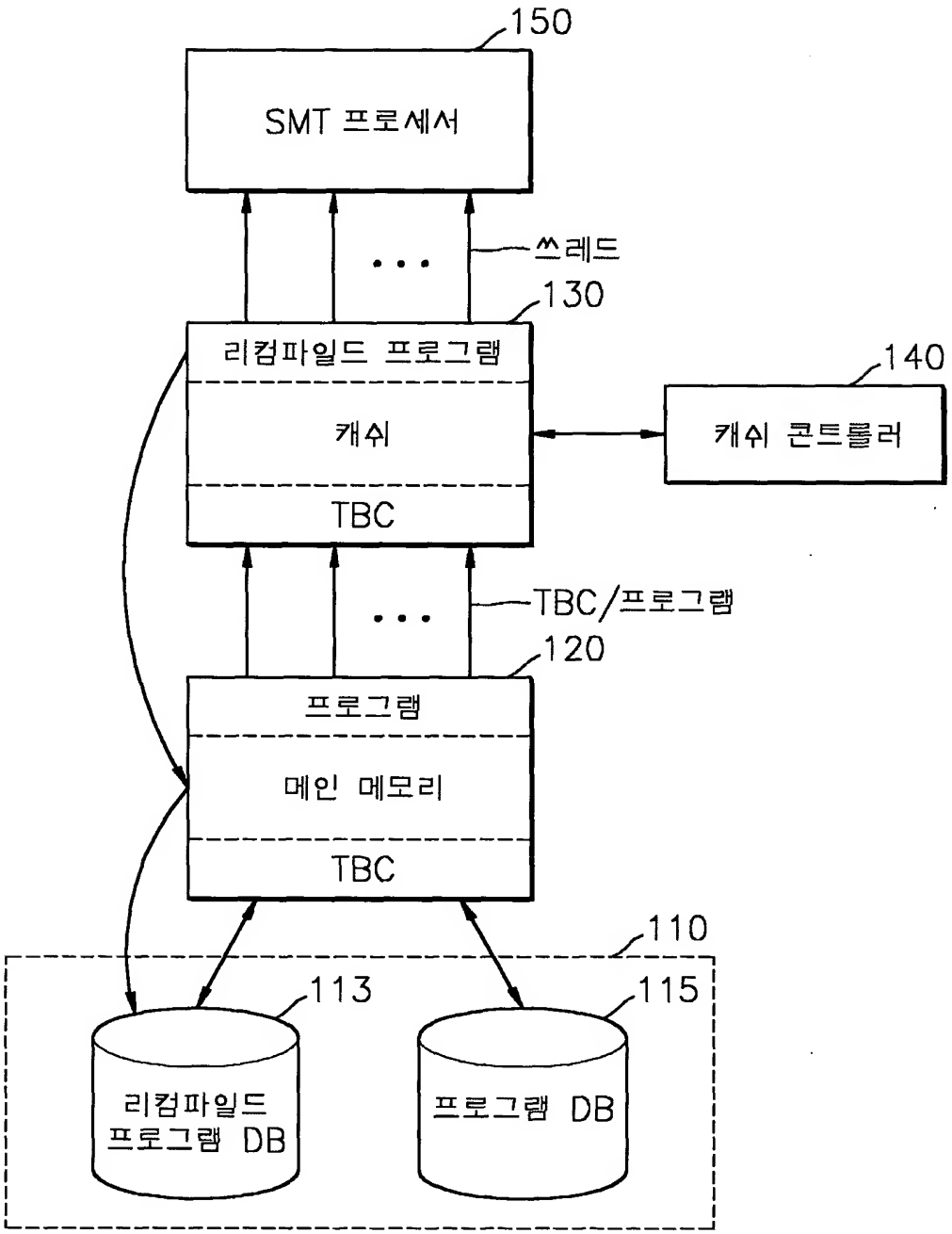
【청구항 20】

제 18항에 있어서, 상기 메인 메모리에 저장된 상기 리컴파일드 프로그램은,

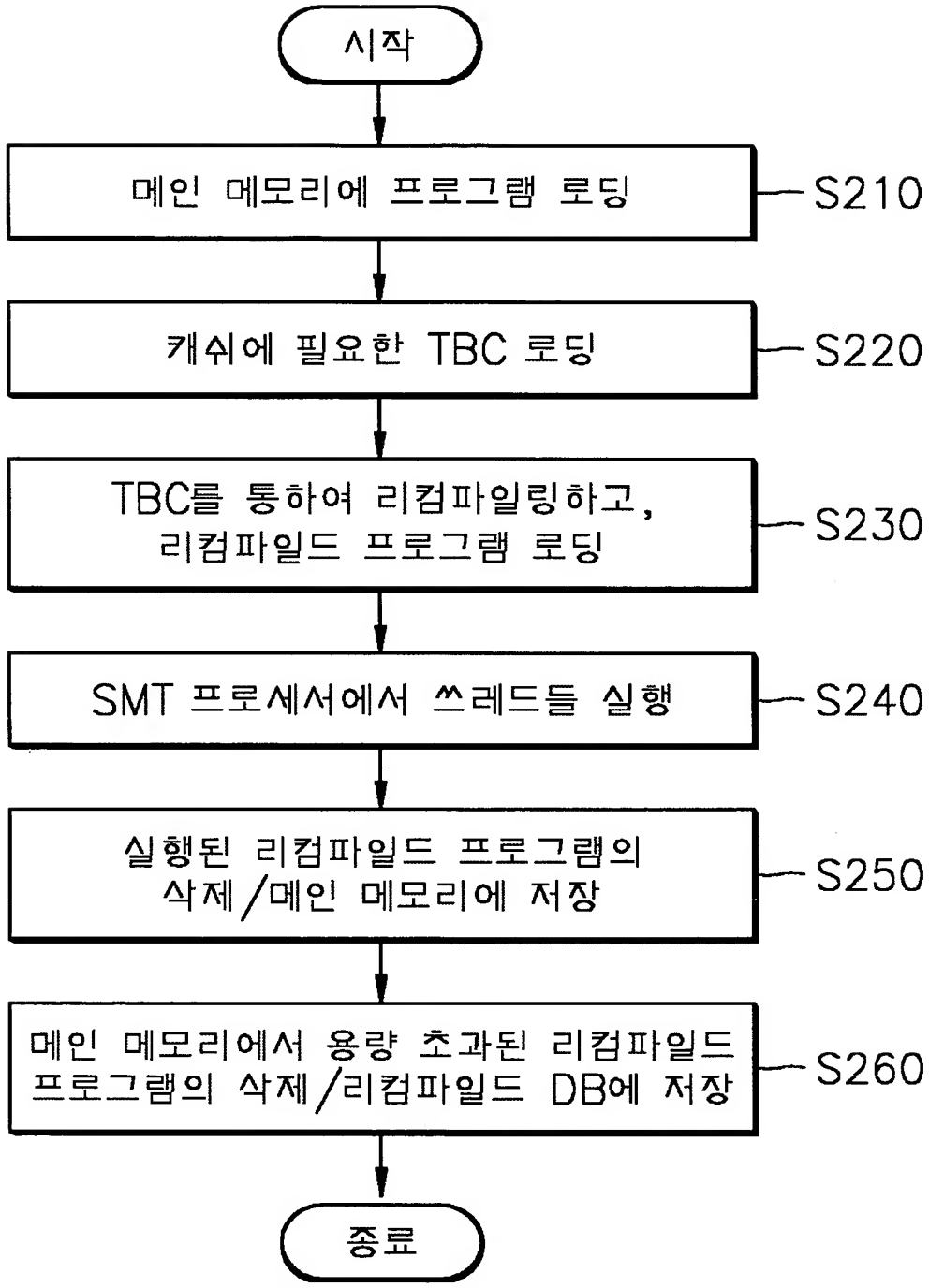
상기 메인 메모리의 용량 초과시 상기 하드 디스크/플래쉬 메모리에 저장되는 것을 특징으로 하는 컴퓨터 시스템의 동시 다중 쓰레딩 방법.

【도면】

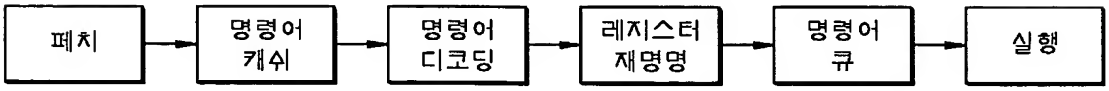
【도 1】


150
SMT 프로세서
쓰레드
130
리컴파일드 프로그램
캐쉬
TBC
140
캐쉬 콘트롤러
TBC/프로그램
120
프로그램
메인 메모리
TBC
110
113
115
리컴파일드
프로그램 DB
프로그램 DB

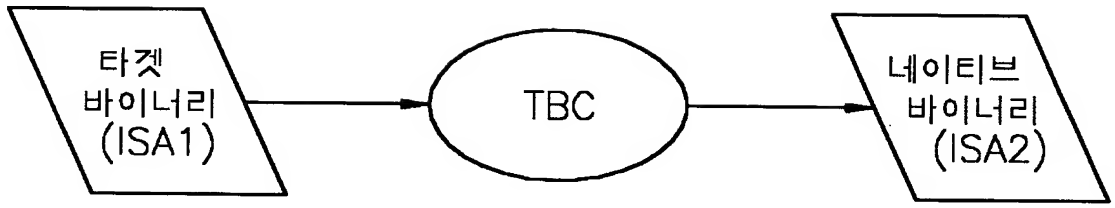
【도 2】


시작
메인 메모리에 프로그램 로딩
S210
캐쉬에 필요한 TBC 로딩
S220
TBC를 통하여 리컴파일링하고,
리컴파일드 프로그램 로딩
S230
SMT 프로세서에서 쓰레드들 실행
S240
실행된 리컴파일드 프로그램의
삭제/메인 메모리에 저장
S250
메인 메모리에서 용량 초과된 리컴파일드
프로그램의 삭제/리컴파일드 DB에 저장
S260
종료

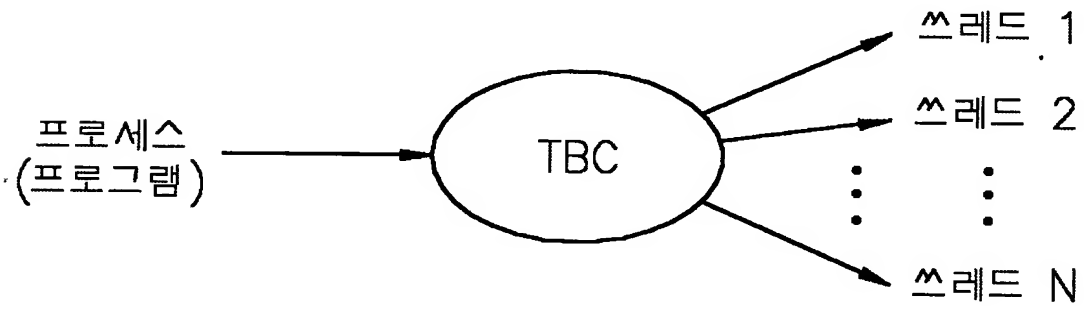
【도 3】


페치
명령어
캐쉬
명령어
디코딩
레지스터
재명명
명령어
큐
실행

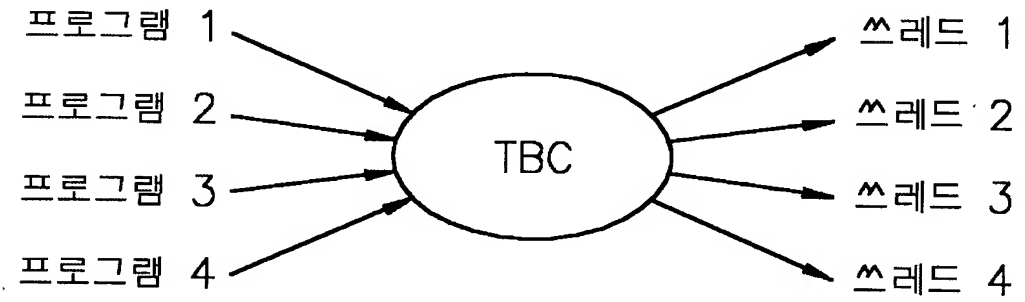
【도 4】


타겟
바이너리
(ISA1)
TBC
네이티브
바이너리
(ISA2)

【도 5】


프로세스
(프로그램)
TBC
쓰레드 1
쓰레드 2
쓰레드 N

【도 6】


프로그램 1
프로그램 2
프로그램 3
프로그램 4
TBC
쓰레드 1
쓰레드 2
쓰레드 3
쓰레드 4